

PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPPPPPPPPPPP		AAAAAAAAAA	TTTTTTTTTTTTTTTT	CCCCCCCCCCCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPP	PPP	AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHH	HHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHHHHHHHHHHHHHHH
PPPPPPPPPPPP		AAA	TTT	CCC	HHHHHHHHHHHHHHHH	HHHHHHHHHHHHHHHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAAAAAAAAAAAAAAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH
PPP		AAA	TTT	CCCCCCCCCCCC	HHH	HHH

```
DDDDDDDD  AAAA  TTTTTTTTTT  BBBB  AAAA  SSSSSSSS
DDDDDDDD  AAAA  TTTTTTTTTT  BBBB  AAAA  SSSSSSSS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DD  DD  AA  AA  TT  BB  BB  AA  AA  SS
DDDDDDDD  AA  AA  TT  BBBB  AA  SSSSSSSS
DDDDDDDD  AA  AA  TT  BBBB  AA  SSSSSSSS
```

....
....
....
....

```
RRRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRRR  EEEEEEEEE  QQQQQQ
RR  RR  EE  QQ  QQ
RR  RR  EE  QQ  QQ
RR  RR  EE  QQ  QQ
RR  RR  EE  QQ  QQ
RRRRRRRR  EEEEEEEEE  QQ  QQ
RRRRRRRR  EEEEEEEEE  QQ  QQ
RR  RR  EE  QQ  QQ
RR  RR  EE  QQ  QQ
RR  RR  EF  QQ  QQ
RR  RR  EL  QQ  QQ
RR  RR  EEEEEEEEE  QQQQ  QQ
RR  RR  EEEEEEEEE  QQQQ  QQ
```

Version: 'V04-000'

```
*****
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*****
```

++

MODULE: DATBAS.REQ

FACILITY: LINKER

ABSTRACT: DATA BASE COMPILE TIME FORMATS

HISTORY:

AUTHOR: T.J. PORTER 01-MAR-77

MODIFICATIONS:

NO.	DATE	PROGRAMMER	PURPOSE
---	---	-----	-----

--

++

FUNCTIONAL DESCRIPTION:

This is a require file that defines the layout (at compile time) of most of the internal data structures of the linker. Symbol table entries are defined separately.

--

Define the layout of and accessing macros for the file descriptor blocks which form a doubly linked list in the order of specification by the user. The FDB contains an RMS auxiliary file name block so that the file may be opened by file id after the first time. The auxiliary file name block contains a descriptor of the resultant file name string (after all logical names and defaults have been applied by RMS on the first open) so that this complete name may be used in error messages and the map. Note however that there is also a descriptor of the name that the user supplied in the command.

```

BYTEBLOCKFIELDS(FDB,
  L_FLINK,4,      | Names are FDB$X_...
  L_BLINK,4,      | Forward link
  L_OMDLST,4,     | Backward link
                  | Listhead for object module descriptors
                  | also used to point to module name list
  W_LIBLSTLNG,2,  | Length of the string which is the
                  | module name list if this is a library
                  | with explicit module extraction
  B_FILFLGS,1,    | File specific flags
  Q_USRNAMDSC,8,   | String descriptor of the user supplied filename
  AC_AUXFNB,NAM$C_BLN); | The RMS auxiliary filename block

```

Define input file flags

GLOBAL LITERAL

```

! **NOTE BIT 0 RESERVED - SET ALWAYS BY CLI**
! ** RE-USED AS FLAG IN LIBRARY SEARCHES**
LNK$S_NEWUDF = 0 : WEAK,      | A module from library added a new undefined symbol to list
LNK$S_LIBR = 1 : WEAK,       | Library flag bit
LNK$S_LIM = 2 : WEAK,        | Linkable image file flag
LNK$S_SELSEAR = 3 : WEAK,    | Selective search file
LNK$S_OPTION = 4 : WEAK,     | Option file
LNK$S_DEBUGGER = 5 : WEAK,   | File contains the debugger
LNK$S_LIBEXTR = 6 : WEAK,    | Explicit module extraction from library
LNK$S_LIBSRCH = 7 : WEAK,    | Library to be searched for undefined symbols

LNK$M_NEWUDF = 1 ^ LNK$S_NEWUDF : WEAK,
LNK$M_LIBR = 1 ^ LNK$S_LIBR : WEAK,
LNK$M_SELSEAR = 1 ^ LNK$S_SELSEAR : WEAK,
LNK$M_DEBUGGER = 1 ^ LNK$S_DEBUGGER : WEAK,
LNK$M_LIBEXTR = 1 ^ LNK$S_LIBEXTR : WEAK,
LNK$M_LIBSRCH = 1 ^ LNK$S_LIBSRCH : WEAK;

```

```

| Make the mask
| For selective search
| File contains the debugger
| Explicit extraction
| Search the library

```

```

: Define offsets into a p-section mapping table (appended
: to module descriptors)

```

```

GLOBAL LITERAL

```

```

PMTSL_PSCDES = 0 : WEAK,      ! Pointer to p-section descriptor
PMTSL_MODCON = 1 : WEAK,      ! Pointer to module contribution data block
PMTSL_SYMLST = PMTSL_MODCON : WEAK, ! Forward list of prematurely defined symbols
PMTSL_SIZE   = 8 : WEAK;      ! Size of an entry

```

```

: Define the layout of an object module
: descriptor and the accessing macros

```

```

LITERAL

```

```

BYTEBLOCKFIELDS(OMD,      ! Initial number of p-sects
L_NXTOMD,4,               ! Names are OMD$X_YY...
L_ALLOC,4,                ! Link to next in file
                        ! Module's contribution to memory
                        ! *** NEXT 2 FIELDS MUST BE CONTIGUOUS
L_MODVBN,4,               ! Virtual block number part
W_BYTOFF,2,               ! And byte offset part of rfa of a library module
B_HIPSCT,1,               ! Highest p-sect number
B_FLAGS,1,                ! Module flags
B_MAPLNG,1,               ! Length of mapping table
B_NAMLNG,1,               ! Name length
T_NAME,SYMS$C_MAXLNG,     ! Name field
AC_PSCMAP,PMTSL_SIZE*NPSECTS); ! P-Section map table

```

```

: Macros to access the RFA of a module

```

```

MACRO

```

```

MODVBN = 0,0,32,0%;      ! Virtual block part
MODBYTE = 4,0,16,0%;     ! Offset within block

```

```

: Object module flags

```

```

GLOBAL LITERAL

```

```

OMD$M_NOPSC = 1 : SHORT WEAK,      ! Set until a p-section is seen
OMD$M_SELSE = LNK$M_SELSE : SHORT WEAK; ! Set if selective search module

```

```

: Define the layout of a program section descriptor
: and the accessing macros

```

```

BYTEBLOCKFIELDS(PSC,      ! Names are PSC$X_YY...
L_FLINK,4,                ! Forward link
L_BLINK,4,                ! Backward link
L_MPCLST,4,               ! Contributing module list
L_SYMLST,4,               ! Owned relocatable symbol list
L_BASE,4,                 ! Base address
L_LENGTH,4,               ! Accumulated (if con)/maximum (if ovr) length
B_ALIGN,1,                ! Alignment of p-section base
W_FLAGS,2,                ! P-Section flags
B_NAMLNG,1,               ! P-Section name length
T_NAME,SYMS$C_MAXLNG);   ! P-Section name

```

```

: Define the layout of a module's p-section contribution data

```



```

: block and macros to access it.
:
: BYTEBLOCKFIELDS(MPC,
:   L_NXTMPC,4,
:   L_OWNOVD,4,
:   L_OFFSET,4,
:   L_LENGTH,4,
:   B_ALIGN,1);
: Names are MPC$X_YY...
: Forward pointer (singly linked list)
: Pointer to module descriptor
: Offset of this contribution from base
: Length of this contribution
: This contribution's alignment
:
: Define the layout of an image section descriptor
:
: BYTEBLOCKFIELDS(ISD,
:   L_NXTISD,4,
:   W_SIZE,2,
:   W_PAGES,2,
:   V_BASVPN,3,
:   B_PAGFCL,1,
:   V_FLAGS,3,
:   B_TYPE,1,
:   L_BASVBN,4,
:   L_IDENT,4,
:   B_NAMLNG,1,
:   T_NAME,SYM$C_MAXLNG);
: Names are ISD$X_YY...
: Singly linked list
: Size of this ISD
: Number of pages in image section
: Base virtual page number
: Page fault cluster size
: I-Sect control flags
: Type code
: Base virtual block number
: I-Sect identification
: Length of name
: I-Sect name.
:
: Define isd flags
:
: GLOBAL LITERAL
:   ISD$M_GBL      = 1 : WEAK,
:   ISD$M_CRF      = 2 : WEAK,
:   ISD$M_DZRO     = 4 : WEAK,
:   ISD$M_WRT      = 8 : WEAK,
:   ISD$V_MATCHCTL = 4 : WEAK,
:   ISD$C_MATNEV   = 0 : WEAK,
:   ISD$C_MATALL   = 1 : WEAK,
:   ISD$C_MATEQU   = 2 : WEAK,
:   ISD$C_MATLEQ   = 3 : WEAK;
:   Global section
:   Copy on reference
:   Demand zero
:   Writable
:   Bit offset to match control field
:   Match never
:   Match always
:   Match equal
:   Match less or equal
:
: Define the layout of the image header's constant data
: record
:
: BYTEBLOCKFIELDS(HDR,
:   W_RECSIZ,2,
:   W_HDRBLKS,2,
:   L_TFRADR1,4,
:   L_TFRADR2,4,
:   W_LIDMAJ,2,
:   W_LIDMIN,2,
:   B_NAMLNG,1,
:   T_NAME,SYM$C_MAXLNG);
: Names are HDR$X_YY...
: Size of this record
: Number of header blocks
: Transfer address 1
: Transfer address 2
: Linker ident major
: Linker ident minor
: Length of image name
: Image name
:
: Some constants of the image header
:
: GLOBAL LITERAL
:   HDR$C_FILLCHR = 255 : BYTLIT WEAK,
:   HDR$C_FILL    = 2 : WEAK;
:   Fill character for header
:   Minimum number of fill bytes
:   ** MUST EQUAL WIDTH OF ISD SIZE F

```

Define the linker version array. Its content is written to image header and checked by the image activator.

```
GENBLOCKFIELDS(LID,      ! Names are LID$X.YY...
MAJOR,2,                ! Major ident (version)
MINOR,2);               ! Minor ident (alteration)
```


0299

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY